The Journal of National Chiao Tung University
Vol. 7, April 1980
pp. 45-50

——短文 Short Notes

論計算機輔助電路分析中選樹的方法
On the Method of Selecting Tree in Computer-Aided Circuit Analysis

蔡　中 Chun Tsai
Department of Electronics Engineering, N. C. T. U.

(Received January 17, 1980)

Abstract — A simple method is described to find a tree in a circuit for computer-aided circuit analysis. This method is based on three rules derived from the properties of tree. It differs from others is that it doesn't use any multiplication operation and obtains the results directly from the incidence matrix.

摘要：本文提出一個簡單的方法為計算機輔助電路分析來選電路的樹。此方法是由樹的特性歸納成三項法則而成。此法有別與他法者是避免用計算機乘的運算並且直接從附帶矩陣上求得。

應用樹（Tree.）的觀念來分析由線性，不與時間變化的元件所組成的電路，演成拓樸網路分析法（Topological Network Analysis）。它是將吾人所欲求得的節電壓（Nodal Voltage）、枝電壓（Branch Voltage），或枝電流（Branch Current）與電源之比值直接與網路的結構相關連的方法。[1][2] 換句話說，用這種方法可以從網路的形態上去求解，毋需先應用克希荷夫電壓定律及其電流定律再加上元件之電性關係式後再逐步求解。但困難的是必需要找到這個電路的所有樹積（Tree Products）及雙樹積（2 − Tree Products），[2] 缺一不可。所以很早就有學者[3]應用計算機快速的運算特性來判斷尋找這些樹與雙樹。不少學者也曾辯論用計算機來求所花費的時間問題。[3][4] 故拓樸法的實際應用尚需研究一種有效的方法。[5]～[7]

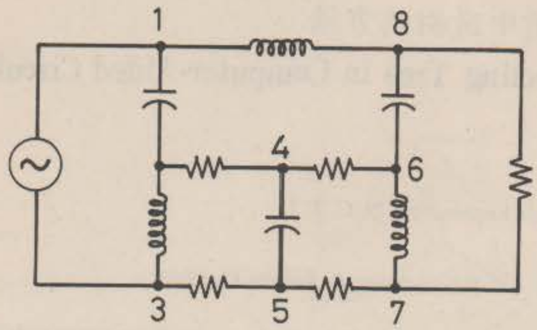
另外一種方法是完全由計算機從電路的形態上建立克希荷夫的二個定律及元件特性來求解。此法雖其求解的程序比拓樸法多，但其簡捷易寫程式的優點而為近代電路分析者所喜用。其原因有：

1. 毋需像在前法（即拓樸法）要不斷地選一組枝（Branch），再判斷它是不是一個樹。如參考4之計算，對圖一的電路共有2187組需要加以判斷。而在後法中，因其只需要一個樹，所以大大地簡化了所需的程式。

2. 在電阻電路或在弦波電源僅求穩態結果的情形時，根本無需找樹，用節點分析法（Nodal Analysis）即可求解。

3. 應用高氏消除法（Gaussian Elimination Method）。

本文所欲探討的是對含有暫態變化的解中計算機選樹時所用的方法。在一般的方法[8]是將簡化的附帶矩陣（Reduced Incident Matrix）變成梯列矩陣（Echelon Matrix）後求得。此法應用了計算機加、減、乘等的運算功能。而在本文內所提出的方法是避免應用計算機乘的運算而直接從附帶矩陣（Incidence Matrix）求得。在下一節內是以一簡單的例子推論出三項判斷的法則，而在第三節內予以應用並加以討論，在最後一節內是個結論與所需的進一步的研究。



圖一

在求含有暫態的解時，對樹內的元件與扣（Cotree ）內的元件都有所要求。對此方面的詳細情況，請參閱參考(9)。在圖二的簡例中以獨立電壓電源、電容器、電阻器、電感器、獨立電流電源的次序來選樹的元件。此電路之附帶矩陣 A_a 為：

$$A_a = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccccc} V & C_1 & C_2 & R_1 & R_2 & L & I \end{array} \\ \left[\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 & -1 \end{array}\right] \end{array} \quad \cdots\cdots\text{式(1)}$$

或其簡化之附帶矩陣 A 為：

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccccc} V & C_1 & C_2 & R_1 & R_2 & L & I \end{array} \\ \left[\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 \end{array}\right] \end{array} \quad \cdots\cdots\text{式(2)}$$

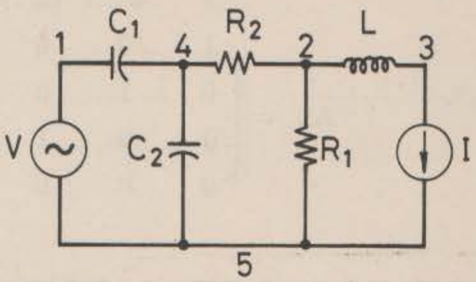
將式(2)轉變成梯列形式（Echelon Form），得：

$$A_{ech} = \begin{array}{c} \begin{array}{ccccccc} V & C_1 & C_2 & R_1 & R_2 & L & I \end{array} \\ \left[\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{array}\right] \end{array} \quad \cdots\cdots\text{式(3)}$$

由式(3)知元件 V、C_1、R_1，與 L 等被選為樹之元件（如欲其詳，請參閱參考 6 第 3 章）。

事實上，吾人可觀察式(1)之附帶矩陣與式(3)之結果得：第一個元件一定會被選上為樹的元件，它的節點（Node ）為 1 與 5；元件C_1連接節點 1 與 4，亦為一選上之元件，因為有一新的節點 4 出現；元件C_2不成樹的元件，因為它沒有新的節點；元件R_1因節點2為新出現者故為樹之一

元件；元件R_2沒有新的節點故不為樹之元件；元件L之節點3為前所未有者，故亦為樹之一元件。此時所有節點均已出現，所以上述之V、C_1、R_1、L為樹之組件，其餘元件則毋需測試了。由是得法則三項（並附簡單的說明或證明）如下：



圖二

法則一：先測試的元件有最大的機會被選為樹之組件，而第一個元件一定為樹之一組件。

說明：因為每一個枝至少有一個樹含之。故先選之枝亦固定樹之形態。

法則二：除第一個樹之元件外，其餘被選上之元件最多也最少有一個在選時為新的節點。

說明：從法則一得到第一個元件的二節點，如另一枝為樹之組件，則必有一節點與舊的節點之一相連，故最多有一個新節點。如二個節點均不是新的，則此元件與已選之元件有環（ Loop ）產生。

法則三：當所有節點均出現後即選定樹之組件。

說明：因為n個節點之電路只有（ n－1 ）個枝之樹。由於第一枝出現二個節點，其餘照法則二，一枝一新節點。故節點均出現時共選（ n－1 ）枝，是為樹之組件。

對上列之三個法則必需要有一個假設，就是應用之電路必需是連接者（ Connected ），無自環枝（ Self-Loop Branch ），並且每個節點都至少有二個枝。對法則二尚需做進一步的說明。如圖二的電路中元件C_1與元件L的位置互換，則其附帶矩陣，簡化之附帶矩陣及其梯列形式分別變成：

$$A'_a = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccccc} V & C_1 & C_2 & R_1 & R_2 & L & I \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & -1 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 & -1 \end{bmatrix} \end{array} \quad \cdots\cdots \text{式(4)}$$

$$A' = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccccc} V & C_1 & C_2 & R_1 & R_2 & L & I \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & -1 & 0 \end{bmatrix} \end{array} \quad \cdots\cdots \text{式(5)}$$

與

$$A'_{ech} = \begin{array}{c} \begin{array}{ccccccc} V & C_1 & C_2 & R_1 & R_2 & L & I \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{bmatrix} \end{array} \quad \cdots\cdots\cdots\cdots 式(6)$$

由觀察知元件C_1之二個節點均為新節點。由於電路是連接的，故它一定為樹之一組件，只是其二個節點中的一個需在其後仍視為一新節點。在程式上如何處理這種情形，將於下節討論。因此，照式(4)知此電路的樹之組件為 V、C_1、C_2及 R_1 等。

按照上節所述的法則，選樹的程式應有的功能如下：

1.對每一節點的情形予以記憶。譬如對尚未選取的節點存 “0”值；而對已選取者（即舊節點）則存入一個非零整數值，如“－1”。故對式(1)之電路，當觀察元件V時，因其為第一個元件，故節點1與節點5均存入數值－1。元件C_1受測時，節點4之值並非員值，為一新節點，所以元件C_1選為樹之一枝而節點4存入－1之值。元件C_2受測時，因二節點均為員值，故C_2不為樹之一枝。如此類推，元件R_1為樹之一枝，節點2存入數值－1。元件R_2不為樹之一枝。元件L為樹之一枝。此時已選上四個枝，達到（n－1）之數，故可停止，不必再選。

2.如遇式(4)的情形，其程序如下。元件V為被選者，其節點1與5均存入數值－1。當元件C_1受測時，因其二節點均為新節點，違反法則二，故暫不選入為樹之一枝。繼續測試元件C_2，因其符合法則二，故選入且節點4存入數值－1。此時因C_2入選，增加了被選之節點，故元件C_1為下一個受測者。結果仍為二新節點。因程式知C_2為一已測過之元件，故測試元件R_1，得其為樹之一元件，節點2成舊節點。此時再回測C_1時，僅有一個新節點，才選入為樹之一枝。故程式需對每一枝亦應有記憶，如尚未測試時為0值；測過為樹之一枝時存以正整數，與測過不為樹之一枝時存以員整數等。

要將本法與前述先求梯列矩陣的方法做一個詳細的比較是很困難的，因為它要涉及程式的寫法與計算機的特性。所以下面是用簡單的例子所作的直接觀察的比較。

以五個節6個枝為例，對下式之電路做個比較：

$$A_a = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & \times & \cdots\cdots & \times \\ 0 & 1 & 0 & 0 & \times & \cdots\cdots & \times \\ 0 & 0 & 1 & 0 & \times & \cdots\cdots & \times \\ 0 & 0 & 0 & 1 & \times & \cdots\cdots & \times \\ -1 & -1 & -1 & -1 & \times & \cdots\cdots & \times \end{bmatrix} \quad \cdots\cdots\cdots\cdots 式(7)$$

此式為形成梯列最簡單的電路，不需做任何改變，計算機用10次比較（或減法）知其已為梯列形式，再用4次比較知前四個元件為樹之組件。但用本法，只要用9次比較即得。再以較複雜的電路為例，像：

$$A_a = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} -1 & 0 & 0 & 0 & \times & \cdots\cdots & \times \\ 1 & -1 & 0 & 0 & \times & \cdots\cdots & \times \\ 0 & 1 & -1 & 0 & \times & \cdots\cdots & \times \\ 0 & 0 & 1 & -1 & \times & \cdots\cdots & \times \\ 0 & 0 & 0 & 1 & \times & \cdots\cdots & \times \end{bmatrix} \quad \cdots\cdots \text{式(8)}$$

則前法除了上述14次減法外尚需用約（10 b－12）次加減與（3 b－3）次乘使之成梯列形式。而用本法時仍只用9次減法即可。本法只在如式(4)時的情形才需要用多於9次的減法。但由於本法毋需用乘法運算而得其顯著的優點。

本文提出一個簡單的方法來選網路的樹。其有別與他法者是完全避免應用計算機乘的運算而直接從附帶矩陣上求得。

然而在分析電路上，在選完樹後，需將簡化之附帶矩陣重組成：

$$A = [A_T \,\vdots\, A_L] \quad \cdots\cdots \text{式(9)}$$

這裏 A_T 是由樹之諸元件組成之小矩陣，A_L 為扣之諸元件所組成者。此時再將A用各種運算法則使 A_T 部份變成對角線為1的矩陣，即得基本分組矩陣（Fundamental Cutset Matrix），以便進行求解的工作。故在前法中可用使A成 A_{ech} 之程式將 A_T 變成梯列後，再將對角線以上的成份用回代法（Back Substitution）使之為零即得。而在本法，如需在選樹之後仍需用成梯列及回代法的程式，則不但增加程式的複雜，而且不合本文的宗旨。故一個避免用計算機乘的運算的程式將式(9)轉成基本分組矩陣的方法正在研究中。

參考文獻

1. S. J. Mason, "Topological Analysis of Linear Non-Reciprocal Networks," Proc. IEEE, Vol. 45, p. 829 ~ 838, June, 1957.
2. S. Seshu and M. B. Reed, "Linear Graphs and Electrical Networks," Addison-Wesley, 1961. Chap. 7.
3. J. Mac Williams, "Topological Network Analysis as a Computer Program," IEEE Trans. on Circuit Theory, Vol. CT-5, p. 228 ~ 229, Sept., 1958.
4. E. W. Hobbs, "Topological Network Analysis as a Computer Program," IEEE Trans. on Circuit Theory, Vol. CT-6, p. 135 ~ 136, March, 1959.
5. W. Mayeda, "Reducing Computation Time in the Analysis of Networks by Digital Computer," IEEE Trans. on Circuit Theory, Vol. CT-6, p. 136 ~ 137, March, 1959.
6. W. K. Chen, "Applied Graph Theory," North-Holland, Ameterdam, 1976. Chap. 5
7. 蔡中，"拓樸分析法之電性意義"，67年度電子材料，元件及電子電路技術研討會，台北，6月23日，第A 52 ~ A 57頁。

8. L. O. Chua and P. M. Lin, "Computer-Aided Analysis of Electronic Circuits," Prentice-Hall, Inc., Englewood Cliff, New Jersey, 1975.
9. 蔡中，"用計算機輔助分析含矽控整流器之電路的方法"，科學發展，第七卷，第十期，第891～897頁，68年10月。